

文字列 数値 判定 方法



正と負の整数をチェックする場合、`-` 記号も含めて、整数全体を表す正規表現にできます。負の符号があってもOK、かつ小数点は含まないという条件で書くには、次のようにすると良いでしょう。

正と負の整数をチェックする正規表現

`-?\\d+` を使います。

- `-?`: マイナス記号 (`-`) を0回または1回許可します。
- `\\d+`: 1つ以上の数字 (0から9) が続くことを指定します。

この正規表現を使って、正と負の整数かどうかをチェックするコードは次のようになります。

```
java

public class Main {
    public static boolean isInteger(String str) {
        return str.matches("-?\\d+");
    }

    public static void main(String[] args) {
        System.out.println(isInteger("123"));           // true (正の整数)
        System.out.println(isInteger("-123"));          // true (負の整数)
        System.out.println(isInteger("0"));             // true (ゼロも整数として許可)
        System.out.println(isInteger("12.34"));         // false (小数は不可)
        System.out.println(isInteger("abc"));           // false (数字以外は不可)
        System.out.println(isInteger("-"));            // false (マイナス記号のみは不可)
    }
}
```

詳細説明

- `"-?\\d+"` は、最初にオプションで`-`を許可し、その後に1つ以上の数字が続く形式にマッチします。
- これにより、`"123"` や `"-123"` は `true` となり、`"12.34"` や `"- "` は `false` になります。

補足

- ゼロも含めて整数として判定されるため、`"0"` も `true` となります。
- マイナス記号のみの文字列（例えば`"- "`だけ）は不正な形式と判定されます。